Research Statement

Di Wang

1 Introduction

My research focus is on algorithm design for large-scale problems with both theoretical significance and practical impacts. With the emergence of extremely large-size datasets and the unprecedented need for understanding the data, machine learning and data mining have become fields where many compelling algorithmic challenges arise. My work seeks to extend results in both continuous and discrete arenas and exploit them in combination.

Combinatorial optimization, especially graph theory, has been instrumental in high performance computing areas such as VLSI design, combinatorial scientific computing and modeling of memory hierarchies. There is a long history in these fields of getting efficiency from combinatorial structures. Given the importance of efficiency in machine learning and data mining, as well as the abundance of combinatorial structures in many applications, methods that make use of these structures have the potential to be far more impactful in modern machine learning applications. In particular, we can exploit classical combinatorial ideas in conjunction with algebraic methods from fields such as continuous optimization and numerical linear algebra to address some of the fundamental drawbacks of numerical methods. Most of my work demonstrates this phenomenon, where I integrate combinatorial and algebraic primitives to design stronger and more practical algorithms that achieve the best of both worlds. This integration is also driving the broader progress at the core of these fields, and I believe the expanding interface between these fields will be the key to progress on many critical modern algorithmic challenges. The approach I take towards designing algorithms, which integrates combinatorial and algebraic perspectives, has consistently led to new insights on such questions, and I plan to further pursue this exciting direction. Main areas of my work include:

1. **Large-scale Optimization**. Iterative first-order (i.e. gradient based) descent methods from (continuous) convex optimization are basic tools that have become workhorses in machine learning for optimizaing over data sets. Building upon recent works on giving provably better bounds for first order methods, I utilized the combinatorial structures of positive linear programs to further speed up the convergence of these algorithms.

2. **Graph Algorithms**. Graphs are of central importance in numerous applications where relational and statistical data arise. In particular, in addition to making technical contributions to algorithms for partitioning, clustering and regression on graphs [9, 20, 14], I recently gave a new tool combining properties of spectral and classical network flow methods which bypasses central bottlenecks for each, which leads to not only stronger provable guarantees, but also improvements on real world datasets in practice [22].

Before proceeding to my work in detail, let's discuss the recent integration of combinatorial and numerical techniques with a more concrete example. A very old idea is to use a spanning tree to pre-condition in the context of iterative Laplacian linear system solving; algebraic techniques give a quick solution on the tree which in turn vastly improves the condition number that limits the convergence of the iterative method. The spanning tree in some sense provides enough global communication across the graph to overcome the limitations of the locality inherent in iterative methods. More recently, this combination of algebra, optimization, and graph theory has lead to dramatically faster algorithms for solving linear systems, and

is being pushed to other problems in optimization including general linear programming. I have a recent result in this framework for optimizing flows under p-norms. To be sure, part and parcel to this progress, was to understand the propagation of information in graphs. Much of my work is in studying these phenomena both in the context of graphs and optimization problems with more complicated interactions.

Examples of my work:

1. On graphs, spectral methods have been a building block in the design of fast graph algorithms both in theory and in practice. Moreover, in the semi-supervised setting such as those used in search engines, some of these methods, e.g. personalized PageRank, are capable of discovering interesting structures while exploring only the local region around a given seed node or set of seed nodes, which makes them very useful in identifying small-scale structures in massive graphs such as the internet or social networks. By combining intuitions of spectral methods with classical ideas from graph algorithms such as network flow, we initiated the study of a flow-based diffusion process which gave distinct information propagation dynamics from spectral methods, and achieved both the stronger optimality guarantee from flow as well as the simple and local properties of spectral methods in the context of graph partitioning. Our diffusion has led to better provable guarantees for several fundamental problems, and we demonstrated improved performance of our diffusion with experiments on real-world graphs [22].

2. I have worked on linear programs with packing and covering constraints, which are classical theory problems and also arise in a wide-range of applications, e.g. data summarization and resource allocation. One essential aspect of applying iterative first-order descent methods is that variables communicate with each other via a fairly local channel, i.e. the gradients. Using insights of the underlying geometries of the specific problems, we tailored discrete techniques and integrated them into first-order methods to smooth the problem space to allow numerical updates according to gradients converge more quickly to the optimum [23, 16].

Current and Future Directions My research draws on techniques broadly from fields such as algorithmic graph theory, convex optimization and numerical linear algebra. The integration of combinatorial and algebraic approaches brings intriguing new perspectives to algorithm design, and I believe this direction can lead to more exciting progress on problems with significant implications both in theory and in practice. Particular future directions I plan to pursue in the near future include

1. More empirically oriented studies and adaptations of our new graph partitioning and local clustering tools in broader domains, such as image processing, information retrieval, ranking, and social network analysis. Also extending the tools to settings such as hyper-graphs, motif patterns, and statistical generative graph models. I intend to work closely with collaborators from the application side on these topics to address more practical questions.

2. The theoretical foundations of the novel flow-based diffusion process on graphs as it should have provably nice properties in broader settings, and can become a primitive in the design of fast graph algorithms both in theory and in practice.

3. The design of complementary combinatorial tools and algebraic routines that can lead to better trade-offs between discrete and numerical approaches to get improved algorithms for more classical problems in algorithmic graph theory, e.g. flow problems in non-linear settings, and more broadly in combinatorial optimization.

Of course the range of interesting problems goes much wider. I plan to push my work further into the intersection of algorithm design and the foundations of machine learning, data mining and network science. The simultaneous emergence of breakthrough techniques in algorithms and a multitude of new problems in these areas is an exciting opportunity.

2 Convex Optimization

Non-negative linear programs (LPs) are LPs where all variables, coefficients and constants are non-negative; two important and well-studied special cases of non-negative LP are packing LP and covering LP. These LPs model many fundamental problems both in theory and in practice, and have long been studied by the optimization and computer science communities. The example of positive LPs highlights a traditional gap between the fields of optimization and computation. Optimization theory studies methods for certain classes of nice functions, e.g., strongly convex or Lipschitz smooth functions, and the performance of these methods is limited by the geometry of the problem space, as well as properties of the objective function. In theory of computation, we design algorithms to optimize specific problems, e.g. maximum matching, and although optimization theory offers abundant intuition on some aspects of our problems, optimization methods may suffer slow convergence as the specific function and problem space we work with can have very poor characterization in terms of the properties necessary for convex optimization methods to perform well. In the case of positive LPs, although first-order iterative methods get optimal dependence on the approximation error ϵ, the convergence rate of these solvers can be limited by the width of the LP, which depends on the largest entry in the LP and its optimal value. On the other hand, there is a line of research in combinatorial optimization, started by Luby and Nisan [15], which gets width-independent solvers but with worse dependence on the ϵ. Together with my collaborators, we adapted the first-order iterative methods with techniques that exploit the problem-specific structures to significantly improve their behavior, and designed positive LP solvers with both width-independence and best dependence on approximation error.

More specifically, we built a discrete lifting technique for covering LPs. This technique reorganizes the geometry of the constraint space of the LPs, so that the optimal solution won't be hidden in ill-shaped corners that are difficult to reach for a particular first-order method developed by Allen-Zhu and Orecchia [2] for packing LPs. We gave a $\tilde{O}(N/\epsilon)$ algorithm for covering LPs, matching a previous breakthrough by Allen-Zhu and Orecchia [2] for packing LPs, where N is the size of the LP and ϵ is the approximation error. In addition, for parallel solvers we designed a methodology that constructs an update vector from the gradient such that the objective function is much smoother along the direction of our update vector than along the actual gradient, so the first-order method can take much larger update steps. Our parallel solver takes $\tilde{O}(1/\epsilon^2)$ time, which is $\Omega(1/\epsilon)$ faster than the previous best algorithm. The methodology developed in our parallel solver has been applied by others to get improved positive SDP solvers [1].

For general non-negative LPs, with my collaborators, I designed a width-independent $\tilde{O}(1/\epsilon^3)$ depth parallel algorithm [16], improving a long-standing bound of $\tilde{O}(1/\epsilon^4)$ [24] for this class of LPs. In particular, we achieved the improvement by integrating insights of first-order iterative methods from continuous optimization and the primal-dual framework from combinatorial optimization. A direct adaptation of our algorithm has been applied by others beyond the case of LPs to develop faster algorithms for submodular functions [8].

3 Local Graph Algorithms

Another main focus of my research is on graph algorithms, especially cut and flow problems in graphs. Graph algorithm is a classical area in combinatorial optimization. Moreover, graphs emerge naturally in practice, e.g. social networks, and are also widely constructed from relational data, e.g. similarity graphs, and thus graph algorithm has become one of the main areas in machine learning and data mining. Most of work is oriented towards modern real-world challenges, which require simple and robust methods that are very efficient when dealing with massive graphs.

One essential aspect of the graph structure that is of central importance in many applications is to understand the bottlenecks in the network, and in particular finding low conductance cuts. Together with my collaborators, I designed a flow-based local method to find low conductance cuts; a local method being one whose running time depends only on the volume of the (smaller side of the) cut that it outputs. While flow and probability mass diffusion (or more generally, spectral methods) have a long history of

competing to provide good graph decompositions [5, 13, 18], local methods are predominantly based on diffusion [4, 7, 3, 25]. Some flow-based local methods [19] have been studied before, but these methods mostly require a given seed set that has large overlap with a low conductance cut, and finding such a seed set relies on diffusion-based methods. Our work is the first primarily flow-based local method for locating low conductance cuts, and has exhibited improved theoretical and empirical behavior over classical spectral methods, e.g. personalized PageRank [3]. So far the tools we built have led to improvements on several fundamental problems in data mining and algorithmic graph theory.

3.1 Local graph clustering

In local graph clustering, we are given one or multiple seed vertices, and want to find a nice cluster (if exists) around the seed node. This task arises broadly in semi-supervised learning and data analysis, e.g. community detection in social networks, and information retrieval on similarity graph constructed from datasets. A local algorithm is desired in this setting, since we don't want a method that needs to touch the entire graph, which can be much larger than the local cluster of interest. Local clustering methods are also used as subroutines in graph decomposition, which in turn plays critical role in many graph algorithms, e.g., the first nearly linear time Laplacian solver by Spielman and Teng [21].

We built upon our local flow algorithm to design a local clustering algorithm []. The key idea is to deviate from prior flow-based methods on how the edge capacities are set, and instead use an adaptive mechanism inspired by local spectral methods to set the edge capacities as the diffusion evolves. This allow us to escape the pitfalls of both spectral and flow on the classical bad examples. In practice, our method is much more robust against noise in the data empirically, and in theory our method is the first local method not subject to the quadratic Cheeger barrier.

3.1.1 Graph decomposition

Graph decomposition algorithms are extensively studied in algorithmic graph theory, and have wide practical applications in unsupervised learning, e.g. social network analysis and image segmentation. A natural bicriteria notion for graph decomposition introduced by Kannan Vempala and Vetta [10], referred to as expander decomposition, is to decompose a graph into *clusters* such that each cluster is richly intraconnected and sparsely connected to the rest of the graph. More formally, given a graph $G = (V, E)$ we aim to find a partitioning of V into $V_1, \ldots, V_k$ for some k, such that the total number of edges across different clusters is small while the conductance of each cluster as an induced subgraph is large. This bicriteria measure is advantageous over other popular measures such as min diameter decomposition, k-center, and k-median since there are simple examples where these measures fail to capture the natural clustering. Moreover, expander decomposition has seen great applications in theory including graph sketching/sparsification, undirected/directed Laplacian solvers, max flow algorithms, approximation algorithms for unique game, and dynamic minimum spanning forest algorithms. With the abundance of massive graphs, it is crucial to design algorithms with running time at most nearly linear in the size of the graph, and thus nearly linear time expander decomposition methods are of great interest both in theory and in practice.

Together with my collaborators, I gave an improved algorithm for this problem that, given an m-edge graph $G = (V, E)$ and a parameter ϕ, partitions V into *expander clusters* $V_1, \ldots, V_k$ for some k, such that the conductance of each clusters is at least $\Omega(\phi)$ while the total number of edges between clusters is at most $\tilde{O}(\phi m)$. Our algorithm runs in $\tilde{O}(m/\phi)$ time. Our result is the first nearly linear time expander decomposition algorithm in the regime of ϕ being at least $1/\log^{O(1)} m$, which is the case for most applications of expander decomposition. Previous methods either runs in super-linear time [17], or provides weaker guarantee in the form that each cluster is only contained in some unknown larger expander rather than being an induced expander [21]. Our algorithm exploits certain dynamic feature of the local flow method, that is, the method remains efficient in the setting where the graph may be updated with edge and node deletions. Not surprisingly, the dynamic feature also led to improved expander pruning algorithms in the dynamic setting, which is the key tool in recent developments of dynamic graph connectivity algorithms.

Moreover, previous algorithms either use approximate sparsest cut or max flow algorithms as a black-box, which has a horrendous polylog$n$ factor in the running time, and is also not easy to implement or even capture with simpler heuristics. In contrast, our result is developed from the first principle and exploits the local flow-based method to find low conductance cut, leading to a simple and basic algorithm of great practical interests.

4 Broader Connections

Deterministic minimum cut algorithm for simple graphs Together with my collaborators, I developed a deterministic algorithm for edge connectivity, i.e. min cut in unweighted graphs, with running time $O(m \log^2 m \log\log^2 m)$ [9]. The result sharpens a previous breakthrough by Kawarabayashi and Thorup which has deterministic running time $O(m \log^{12} m)$ [12], and is even better than the fastest randomized algorithm with running time $O(m \log^3 m)$ by Karger [11], although Karger's result generalizes to weighted graphs. This work is built upon our local flow-based tool to find low conductance cuts.

Graph Sparsification As a consequence of our stronger expander decomposition result, we also achieved better graph sparsification. Beyond technical improvement on the running time and being simpler to apply, the stronger guarantee that each cluster is an induced expander greatly extends the range of applications of graph sparsification to cases where previous approaches are not sufficient, e.g. short-cycle decomposition and dynamic graph connectivity, and multi-objective preserving sparsification.

p-norm flows Together with my collaborators, I recently had a result for solving a large class of flow problems on unit weighted graphs to high accuracy in almost-linear time, where the objective functions interpolate between ℓ_2 (i.e. electrical flow) and ℓ_∞ (i.e. max flow) norms. This is the first almost linear time flow algorithm beyond the case of $p = 1$ (i.e. shortest path) and $p = 2$ (i.e. electrical flow). Our framework is inspired by the routing based solver for Laplacian linear systems by Spielman and Teng [21], and greatly extends the technical tools in their work to much broader settings. One of the key components is the stronger expander decomposition result describe earlier to preserve multiple objectives under sampling. The applications of our result include: (1) a new approach for computing approximate max flows on undirected graphs that does not use oblivious routing; (2) the first almost-linear time high-accuracy algorithms for p-norm semi-supervised learning on graphs when $p < 2$; and (3) the first almost-linear time algorithm for approximate total variation minimization.

References

[1] Zeyuan Allen-Zhu, Yin Tat Lee, and Lorenzo Orecchia. Using optimization to obtain a width-independent, parallel, simpler, and faster positive sdp solver. In *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 1824–1831, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics.

[2] Zeyuan Allen Zhu and Lorenzo Orecchia. Using optimization to break the epsilon barrier: A faster and simpler width-independent algorithm for solving positive linear programs in parallel. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1439–1456. SIAM, 2015.

[3] Reid Andersen, Fan R. K. Chung, and Kevin J. Lang. Local partitioning for directed graphs using pagerank. *Internet Mathematics*, 5(1):3–22, 2008.

[4] Reid Andersen and Yuval Peres. Finding sparse cuts locally using evolving sets. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 235–244. ACM, 2009.

[5] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2), 2009.

[6] Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors. *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[7] Fan R. K. Chung. A local graph partitioning algorithm using heat kernel pagerank. *Internet Mathematics*, 6(3):315–330, 2009.

[8] Alina Ene, Huy L. Nguyen, and Adrian Vladu. Submodular maximization with packing constraints in parallel. *CoRR*, abs/1808.09987, 2018.

[9] Monika Henzinger, Satish Rao, and Di Wang. Local flow partitioning for faster edge connectivity. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1919–1938. SIAM, 2017.

[10] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings - good, bad and spectral. In *FOCS*, pages 367–377. IEEE Computer Society, 2000.

[11] David R. Karger. Minimum cuts in near-linear time. *J. ACM*, 47(1):46–76, 2000.

[12] Ken-ichi Kawarabayashi and Mikkel Thorup. Deterministic global minimum cut of a simple graph in near-linear time. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 665–674. ACM, 2015.

[13] Rohit Khandekar, Satish Rao, and Umesh V. Vazirani. Graph partitioning using single commodity flows. *J. ACM*, 56(4), 2009.

[14] Rasmus Kyng, Richard Peng, Sushant Sachdeva, and Di Wang. Flows in almost linear time via adaptive preconditioning. 2018. Recently submitted.

[15] Michael Luby and Noam Nisan. A parallel approximation algorithm for positive linear programming. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pages 448–457, New York, NY, USA, 1993. ACM.

[16] Michael W. Mahoney, Satish Rao, Di Wang, and Peng Zhang. Approximating the solution to mixed packing and covering lps in parallel $\tilde{O}(epsilon^{-3})$ time. In Chatzigiannakis et al. [6], pages 52:1–52:14.

[17] Danupon Nanongkai, Thatchaphol Saranurak, and Christian Wulff-Nilsen. Dynamic minimum spanning forest with subpolynomial worst-case update time. In *FOCS*, pages 950–961. IEEE Computer Society, 2017.

[18] Lorenzo Orecchia, Leonard J. Schulman, Umesh V. Vazirani, and Nisheeth K. Vishnoi. On partitioning graphs via single commodity flows. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 461–470. ACM, 2008.

[19] Lorenzo Orecchia and Zeyuan Allen Zhu. Flow-based algorithms for local graph clustering. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1267–1286, 2014.

[20] Thatchaphol Saranurak and Di Wang. Expander decomposition and pruning: Faster, stronger, and simpler, 2018. To appear at SODA 2019.

[21] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 81–90. ACM, 2004.

[22] Di Wang, Kimon Fountoulakis, Monika Henzinger, Michael W. Mahoney, and Satish Rao. Capacity releasing diffusion for speed and locality. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3598–3607. PMLR, 2017.

[23] Di Wang, Satish Rao, and Michael W. Mahoney. Unified acceleration method for packing and covering problems via diameter reduction. In Chatzigiannakis et al. [6], pages 50:1–50:13.

[24] Neal E. Young. Sequential and parallel algorithms for mixed packing and covering. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 538–546. IEEE Computer Society, 2001.

[25] Zeyuan Allen Zhu, Silvio Lattanzi, and Vahab S. Mirrokni. A local algorithm for finding well-connected clusters. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Proceedings*, pages 396–404. JMLR.org, 2013.